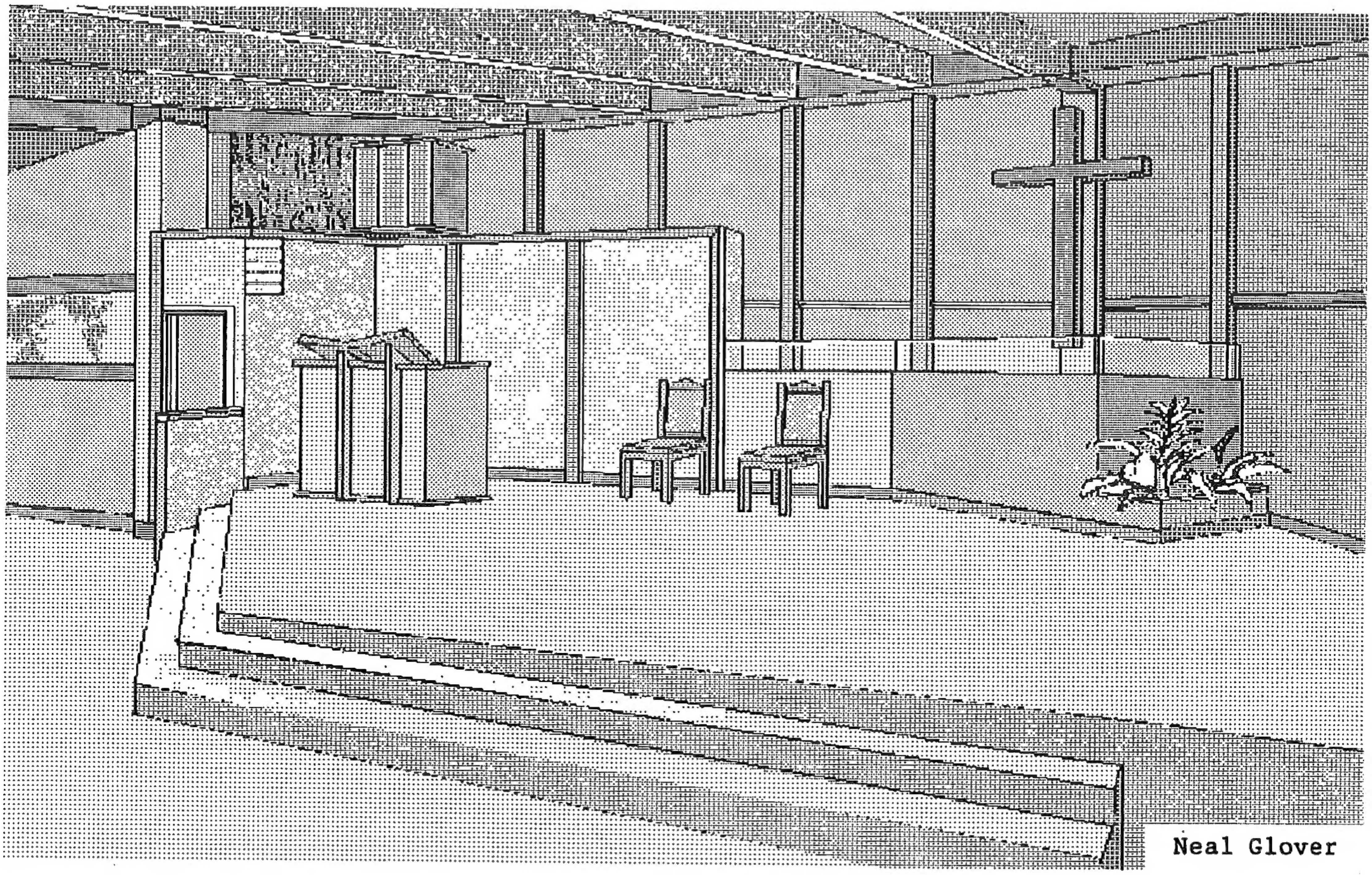
AMIGA

WORKBENCH

$2 RRP

Registered by Australia Post — Publication No. VBG7930

Number 23 Circulation: 1100 April 1988



Neal Glover

Yes, we're back to Sunday meetings!

Next AUG Meeting

Sunday, April 17th, 1988 at 2pm

(Doors open at 1pm, meeting starts at 2pm sharp)

AUG meetings are held in the Rotunda at Monash University
Wellington Road, Clayton Melways map 70 reference F10 and map 84A

Amiga Users Group inc, PO Box 48, Boronia, 3155, Victoria, Australia

Australia's Largest Independent Association of Amiga Owners
The *Amiga Users Group Inc* has no affiliation with Commodore
AMIGA is a trademark of Commodore–Amiga, Inc

Who Are We?

The Amiga Users Group is a non-profit association of people interested in the Amiga computer and related topics. With almost 900 members, we are the largest independent association of Amiga users in Australia.

Club Meetings

Club meetings are held at 2pm on the third Sunday of each month in the Rotunda at Monash University, Wellington Road, Clayton. . Details on how to get there are on the back cover of this newsletter. The dates of upcoming meetings are:

Sunday, April 17th at 2pm
Sunday, May 15th at 2pm
Sunday, June 19th at 2pm

Production Credits

This month's newsletter was edited by Peter Jetson. Equipment and software used was: TurboDOS S-100 computer, Brother HR-40 printer, Gemini 10x printer, Wordstar, Fancy Font and Grabbit.

Copyright and Reprint Privileges

Contributions

Articles, papers, letters, drawings and cartoons are actively sought for publication in Amiga Workbench. Please submit your contributions on disk, since that means they don't have to be re-typed! All disks will be returned! Please save your article in **text-only** format (If it can be loaded by ED, it is text-only). Absolute deadline for articles is 16 days before the meeting date. Contributions can be sent to: The Editor, AUG, PO Box 48, Boronia, 3155.

Membership and Subscriptions

Membership of the Amiga Users Group is available for an annual fee of $20. To become a member of AUG, fill in the membership form in this issue (or a photocopy of it), and send it with a cheque for $20 to:

Amiga Users Group, PO Box 48, Boronia, 3155

Public Domain Software

Disks from our public domain library are available on quality 3.5" disks for $8 each including postage on AUG supplied disks, or $2 each on your own disks. The group currently holds over 170 volumes, mostly sourced from the USA, with more on the way each month. Details of latest releases are printed in this newsletter, and a catalog disk is available.

Member's Discounts

The **Amiga Users Group** negotiates discounts for its members on hardware, software and books.

Currently, **Technical Books** in Swanston Street in the city offers **AUG** members a 10% discount on computer related books, as does **McGills** in Elizabeth Street. Just show your membership card. Although we have no formal arrangements with other companies yet, most seem willing to offer a discount to AUG members. It always pays to ask!

Back Issues of Newsletter

All back issues of Amiga Workbench are now available, for $2 each including postage. Back Issues are also available at meetings.

AmigaLink - Our Bulletin Board System

The Amiga Users Group operates a bulletin board system devoted to the Amiga, using the Opus message and conferencing system. AmigaLink is available 24 hours a day on (03) 792 3918, and can be accessed at V21 (300bps), V22 (1200bps) or V23 (1200/75bps), using 8 data bits, 1 stop bit and no parity.

AmigaLink is part of the world-wide Fido/Opus network of bulletin boards, and we participate in the national and international Amiga conferences. AmigaLink has selected Public Domain software available for downloading, and encourages the uploading of useful public domain programs from its users. AmigaLink is FidoNet node number 631/324.

Newsletter Advertising

The Amiga Users Group accepts commercial advertising in Amiga Workbench subject to the availability of space at these rates:

Quarter page	$20
Half page	$40
Full page	$70
Double page spread	$120

These rates are for full-size camera-ready copy **only.** We have no photographic or typesetting facilities. Absolute deadline for copy is 16 days before the meeting date. Send the copy and your cheque to: The Editor, AUG, PO Box 48, Boronia, 3155, Victoria.

Amiga Users Group Committee

Co-ordinator:	Bob Scarfe	376 4143 Kensington
Vice Co-ord:	Fergus Bailey	211 7845 Malvern
Meeting Chair:	Ron Wail	878 8428 Blackburn
Secretary:	John Elston	375 4142 M' Ponds
Treasurer:	(temporarily vacant)	
Membership:	Neil Murray	794 5683 Dandenong
Purchasing:	Bohdan Ferens	792 1138 Dandenong
Book Library:	Joan Wood	580 7463 Aspendale
Disk Library:	Geoff Sheil	578 8362 Brighton
	Margaret Bedson	578 8362 Brighton
Editor:	Peter Jetson	762 1386 Boronia

1988 CEBIT Show Announcements

by Lauren Brown

[Editor's Note: The following is from a message left on a US computer network by Lauren Brown of CATS (Commodore Amiga Technical Support) on March 17th, 1988]

With over 500 members of the press in attendance, Commodore held a press conference the first day of CEBIT (16-Mar-88). Chairman Irving Gould took this opportunity to announce the continuation of the company's successful financial picture. For the 6 month period ending 31-Dec-87, net income rose to $34 million, up from $25.5 million in the same period one year ago. This represents an increase of 33%.

Mr Gould stated that 80% of the net income came from products not present in the Commodore line 2 years ago. Mr Gould announced the redesign of the PC line, and the further development of the Amiga computer family. Plans for the Amiga 2500 and Amiga 3000 were discussed. Mr Gould stressed that "the A2000 will remain compatible with all new attributes of new Amiga products". Details of new products are:

The AMIGA 2500AT, which will have the following specifications:

* 68020/68881
* 80286 Bridgeboard
* 40 Megabyte autoboot hard disk drive

The AMIGA 2500UX, which will be equipped with:

* 68020/68881
* Autoboot Hardrive (with option of 100 MG harddrive)
* Unix 5.3 compatible operating system

The Unix included with the A2500UX is a proprietary high performance windowing user shell. Also optional for the A2500UX will be a tapestream storage device.

It was also announced that Commodore is continuing work on the Amiga 3000, which will be based on Motorola's 68030 chip. Some of the attributes of this machine include networking capabilities and multiple mass storage devices of up to 2.2 gigabytes. Release of the A3000 is slated for calendar year 1989. Other new products also shown at the Commodore booth include:

A2620 - 68020/68881/68851 coprocessor for the A2000 with 2-4 Megabytes of RAM

A2024 - High Resolution Gray Scale Amiga Monitor

A2300 - Amiga 2000 Genlock

A2286 - 80286 based Amiga bridgeboard

Transputer project

In other news from the press conference, Commodore announced a cooperative effort with a large scale research group to develop a high performance workstation. The West German Society for Biotechnological Research and Commodore are working together to develop the workstation which will be based on the Amiga 20000 with Transputer. The workstation is aimed at the chemical, pharmaceutical and food industries, and is considered ideal for graphics applications and display of molecular structures.

Chairman Gould announced no plans for Commodore to increase its computer prices due to DRAM price increases. We will continue to supply our market at price levels now in effect. It was also announced that enhanced graphics chips with non-interlace mode will be available for the Amiga 500 and Amiga 2000.

Commodore is the leader in the West German' microcomputer industry, with a market share of over 50%. With professional systems, Commodore ranks in 2nd place in Germany, and is unchallenged in the home market. European press reaction to the conference has been very positive. Some of the headlines from the show include: "Commodore and Nixdorf Grow, Grow, Grow" and "Commodore Attacks the Professional and Business Market".

Commodore and, more especially, CATS, want to thank all developers for their ongoing support. It is this support which has made the advancements possible.

Hoping to see all of you in Washington - Lauren

==

Fire Power - An Update

by Stan Thomas

In these pages last month, I wrote a review of 'Fire Power' from Microillusions. In the review, I stated that on my system Fire Power crashed due to a corrupted memory list every time I ran it. Well, things have changed. Soon after the article was submitted, I mentioned the problem in passing to another customer in the city branch of Ultraphase Computers. I said that my current system setup is an Amiga 1000 with 512K ram and a 1010 external drive. "Try disconnecting the external drive", he said. It was some time before I was able to try this, and quick test revealed that taking off the second drive released about 23K of extra ram. Anyway, the end result is that the game now works perfectly.

Since my playing time is no longer restricted to 15 minutes, my estimate of this game has gone up somewhat. I can get much more involved in the gameplay, and I now recommend it, providing you are prepared to disconnect your second drive if the Guru visits (with Exec Library Alert Code 81000005).

The documentation provided with this game is nearly zero - you read it when you load it - and no mention was made that disconnecting external drives may be necessary. An Amiga 1000 with 512K and an external drive must surely be a very common hardware setup, and if proper testing using this configuration revealed this fault, it should have been made clear somewhere in the packaging that external drives may have to be disconnected.

68000 Machine Code for Basic Programmers

If you are one of those few people who like your computer to jump to your commands and not someone elses (ie if you actually like to program), then you might like to know that it is possible to write your machine code in Basic first and then convert it down to machine code later.

Why bother with machine code? Well, it is worth remembering that the knowledge required to write machine code can be useful even if you usually program in another language. If your routine consists of a simple piece of code repeated many times (such as searching, sorting and graphics), the difference in speed can be quite spectacular.

It does take more time to write in Assembly (the language used to write machine code) because the statements are rudimentary - not because they are more difficult than Basic.

A line to line correspondence between Basic and Assembly can often be achieved by using a restricted set of Basic statements. This way you can have an easy to read program which can be developed quickly and then converted to machine code with a reasonable expectation that it will work.

The 68000 CPU inside your Amiga has 8 data variables d0 to d7 and 8 address variables a0 to a7. These are long integers. All other data is stored in the memory.

It is important that you learn to PEEK and POKE the memory as these instructions have a machine code equivalent.

```
d0 = PEEK(a0)   becomes   d0 = M(a0)
POKE a0, d0     becomes   M(a0) = d0
```

In the above, and the table below, I have used M() to denote the computers memory in demonstrating the addressing modes.

Consider the following:

```
d0 = d1
d0 = a0
d0 = M(a0)
d0 = M(a0) : a0 = a0 + 1
a0 = a0 - 1 : d0 = M(a0)
d0 = M(a0 + 14)
d0 = M(a0 + d2 + 45)
d0 = M(642)
d0 = 89
```

If you can understand the above statements, then congratulations - you now know the addressing modes of the 68000 cpu. Easy, wasn't it?

The computer's memory is just a list of locations, where each location can hold a number between 0 and 255. A string is stored in the computers memory as a list of numbers (the ASC code of each character). It uses one location for each character. See Amiga Basic Manual A-1 for asc codes.

Now let's write a program. This is my version of the subroutine given in the Amiga Basic manual 6-11, that converts a lower case string to an upper case string. It works by PEEKing each number and testing to see if it is between 97 and 122 (ie a to z). If it is, then it subtracts 32 to convert the character to upper case. It then POKEs the number back into the memory.

The SADD routine returns the String ADDress of the location in which the first character is stored.

The LEN returns the LENgth of the string.

```
INPUT s$
GOSUB Ucase
PRINT s$

Ucase:
 d0 = LEN(s$)
 a0 = SADD(s$)
 d1 = 0
loop:
 d1 = PEEK(a0)
 IF d1 < ASC("a") THEN skip
 IF d1 > ASC("z") THEN skip
 d1 = d1 AND 223
skip:
 POKE(a0),d1 : a0 = a0 + 1
 d0 = d0 - 1
 IF d0 <> 0 THEN loop
 RETURN
```

You will notice that 32 is subtracted from d1 by ANDing with 223. This is an example of how the knowledge needed for machine code is useful even if you use a higher level language. You may substitute the line with d1 = d1 - 32.

Below is the Assembly version, along with the Basic version in the comment column, to show the one to one correspondence possible.

```
Ucase:
' save registers used- not relevant to Basic
    movem.l   a0/d0-d1,-(sp)

                               ' Basic Version
    move.l    16(sp),d0        ' d0 = LEN(s$)
    move.l    20(sp),a0        ' a0 = SADD(s$)
    clr.l     d1               ' d1 = 0
loop:
    move.b    (a0),d1          ' d1 = PEEK(a0)
    cmp.b     #'a',d1          ' IF d1 < ASC("a")
    blt       skip             ' THEN skip
    cmp.b     #'z',d1          ' IF d1 > ASC("z")
    bgt       skip             ' THEN skip
    and.b     #223,d1          ' d1 = d1 AND 223
skip:
    move.b    d1,(a0)+         ' POKEa0, d1 : a0 = a0 + 1
    subq      #1,d0            ' d0 = d0 - 1
    bne       loop             ' IF d0 <> 0 THEN loop

' restore registers not relevant to Basic

    movem.l   (sp)+,a0/d0-d1

    rts                        ' RETURN
```

Now I'll elucidate some of the 68000 assembly language mnemonics I've used, since there is no exact equivalence, only similiarity.

move - a LET without any expressions
clr - CLeaRs the variable to zero
cmp - CoMPares two values. Used to set flags for a conditional jump
blt - Branch if Less Than
bgt - Branch if Greater Than
and - same as AND in Basic
subq - subtracts a number between 0-7 from reg.
bne - Branch if Not Equal
rts - RETURN

.b BYTE number in one location, like PEEK
.w WORD (short integer) 2 locations, like PEEKW
.l LONG (long integer) 4 locations, like PEEKL

The above Assembly is converted into machine code (which is a list of short integers). It can then be called by giving Ucase the address of the first number in the list. In the example at the top of the next column, the subroutine is in a set of DATA statements. However in practice, they might be saved and loaded from disk as a file.

The routine does the same thing as the UCASE$ in Amiga Basic and was given only as an easy to understand routine for demonstration purposes.

```
DIM code%(21)
FOR i = 0 to 21
     READ code%(i)
NEXT

INPUT s$

length&  =  LEN(s$)
addr&  =  SADD(s$)

Ucase  =  VARPTR(code%(0))

CALL Ucase(length&, addr&)

PRINT s$

DATA &H48E7,&HC080,&H202F,&H0010,&H206F,&H0014
DATA &H4281,&H1210,&H0C01,&H0061,&H6D0A,&H0C01
DATA &H007A,&H6E04,&H0201,&H00DF,&H10C1,&H5300
DATA &H66E8,&H4CDF,&H0103,&H4E75
```

Graphic Users SIG

Theo den Brinker

The Graphic Users Special Interest Group (GUSIG) is for Amiga users interested in creating visual images on their machines using existing software. This can be either commercial or public domain. The February meeting was a trifle disorganised, but future meetings should have more direction.

Displayed at the meeting were some of my images which I had copied from paintings originally done in the 1930's. Copying existing paintings, while not an original activity, does provide a good starting point in terms of learning about style, colour and form from recognised artists. It also provides lessons in the strengths and weaknesses of using the Amiga artistically. Besides, I'm bored with mirrored balls bouncing on checkerboards and their derivatives being seen as the highpoint in the Amiga's graphics!

Also shown were two image processors, **Butcher** and **FPic**. **Butcher** is available commercially while **FPic** is on the public domain disk **Amicus 20** under the title of **Imageproc**. **Fpic** was designed for the processing of digitized black and white photographs, but it also creates some interesting effects on colour images. The functions include three edge detection routines, pixalize, sharpen and binary. Documentation describing the commands is given along with two black and white photos you can experiment with. **FPic** works from either the CLI or Workbench. **Amicus 20** also has three programs for those who want to create image directly with the graphic elements provide by the Amiga:

BobEd is for the creation of BOBs, sprites or any other small graphic objects.

Spritemaster II allows all sorts of manipulations of sprites.

Blitlab when you want to find out about the Blitter.

These activities are a bit beyond me but for those who are interested all three programs look to have good user interfaces and are accessable from the Workbench. (While Workbench accessability for many programs is not strictly necessary it does show, to me at least, a positive attitude in creating programs in the Amiga style)

If you have any questions ring me on 509 1050 (home) or 819 6088 (work). Theo den Brinker.

==

Aegis Impact
Business Graphics for the Amiga

by Michael Hassett

Computer graphics for business applications has become an important use for the computer in recent years. Applications might be either the direct presentation of the images produced "on screen", in printed form, or output in a photographic or video mode.

Impact allows for the creation of graphs combined with text and images. Bar, Pie and Line graphs can be created in 2D or 3D by either the input of statistics, or drawing using Impact's range of tools and editing facilities.

When Impact is first loaded you will be presented with a blank screen, a fast menu and the familiar pulldown menus. A reasonable degree of resolution is available as it operates in 640 x 200, or medium resolution, mode. To create a graph, you need to enter a range of statistics into the graph builder. An example of this would be, for example, the comparison of computer sales over a three year period:

	1988	1989	1990
Amiga	1000	2000	3000
Apple	1000	1050	900
IBM	950	900	500

Firstly, you are be asked to enter the "series", in this case we will make the years the series. A table will then be presented requiring the data to be entered firstly for 1988, these would be the figures in the above table for the individual computers, you then create a new table for the values of 1989 and for 1990.

There is now enough information to build a graph. At the bottom of left of the screen, you are given the option of drawing or plotting.

The basic difference between these two options are that with plot you cannot edit the image but with draw you have access to the host of editing features.

So for this case as an example we will select draw. Moments later, we get a fully labelled bar graph in glorious colour. From here the abilities of Impact are enormous - simply by selecting from the menu we can change the graph to a pie or line chart and in 3D just as quickly. It is your imagination that puts the limits on the finished product from here. You get a choice of background grids, colours, patterns, about 60 fonts and the ability to cut, paste, move or merge any section of the graph.

Once the graph is completed, you can save it as a slide, a graph, a window or a table depending upon whether you wish to re-use the data to create another version; transfer it to another graphics package for further embellishment, then transfer to video or 35mm slide; or have it saved as an Impact Slide. When saved as a slide you are able to utilise Impact's slide program which very rapidly allows you to build your own slideshow on screen.

Aegis Impact has been well constructed and is quick and easy to use. The package looks as though is has been intended for professional use, but it would need to operate in 640 x 400 to achieve a professional appearance. Also, transferring the output to video tape is let down by the poor quality of the Amiga's composite video output. Devices such as Polaroid Palette are now available for transfer to 35mm slide or prints.

This is one of the serious pieces of software available for the Amiga, and as a first release, its creators should be congratulated. I look forward to seeing the next release and would expect it to incorporate a Hi-Res Mode as well as the ability to bring IFF files into the package.

As it stands, Impact is an excellent product for the production of "in-house" presentations or for developing ideas for graphical layouts, and the range of fonts and the ability to manipulate them are worth looking at, particularly if you are not happy with the way "Images" and "Paint" handle text.

==

COMING OUT OF THE CLOSET

by Willie C. de Lyte

Real men don't eat quiche, nor do they write BASIC programs. Assembly language is their realm, as are meat pies with sauce. Such contentious myths need to be exposed for what they really are, and I will attempt in this article to defend the honor of a particular strain of BASIC, namely True BASIC, and I might even have time to prove that I'm a real man, despite the fact that I like quiche.

I will attempt to describe True BASIC to you as dispassionately as possible, but as you well know - being Amiga owners - it is difficult not to be enthusiastic about a good product.

I will firstly deal with an obvious question: why would one want to buy True BASIC when AmigaBASIC comes free with the machine? Not an easy question to answer, really, in view of the fact that money is almost everything. Perhaps it will help if I describe the process by which I came to buy, and ultimately like, True BASIC.

First, a little background, so that you may better understand my attitudes. My initiation into computing was at University, where I cut my teeth on a good smattering of languages, from micro assembler, through FORTRAN (god forbid!) and ALGOL, to such arcane masterpieces as LISP and SIMULA. But over the next ten years, during the times I managed to hold down a regular job, my programming gave way to what is euphemistically called "Systems Programming", an amorphous title that describes a plethora of tasks ranging from token assembler programming to patching shonky operating systems. It was during this spell that I came into contact with commercial programming with PL/1 and, yuk, COBOL. Life was pretty dull, except for the advent of the SAS language, but that's another story. I only mention my dubious past to try to assure you that I'm not afraid of unfamiliar languages, that I can use them, and that I have enough technical experience to lend a little credibility to my story.

Back to the subject: programming on the Amiga. I tried my hand at "C". It was a bit of a shock at first, with so many minute or obscure symbols scattered about the place. But I came to grips with that. Then I tried Modula II. That was nicer. And I tried assembler. I love assembler, but it's a tedious language to use, especially when you want to knock up a quick program to perform a once-off task. Similarly, "C" and Modula II are tedious, albeit much less so than assembler. The essential thing I dislike is the time it takes to perform the "code-inspect-compile-link-crash-debug-recode" cycle. It drives to me to despair! Oh, sure, if you've been programming with these languages on the Amiga long enough, you'll have stock-standard blocks of code tucked away for stock-standard areas of endeavour such as screens, windows, serial and parallel ports, the audio device, and so on; but piecing it together, and building the body of the program, is still subject to that vicious cycle. What I was after was economy of effort, whereby one could imagine an application, and with little effort test the idea. Also, when it comes to computing, I'm lazy, very lazy, and with the little time I have to devote to personal computing, every second counts.

AmigaBASIC is at the right end of the comfort spectrum, allowing a programmer to try out ideas in an instant. Well, that's how it would be, ideally, but sadly it did not appear to be the case with the copy of AmigaBASIC I had. The interface is a DOG. I've never seen such painfully slow scrolling in an editor. Yes, yes, I know it checks for keywords and all that, but it does a helluvalot less scrutinising than WordPerfect, and I don't see WordPerfect in agony every time it scrolls. The menu options are scarce, and leave one exasperated at times. And what about its response when you have a syntax error! It pops up a very sleepy requestor, which seems to want to stay around for the life of your program, and once you have dispensed with it, AmigaBASIC searches fruitlessly for the error. Some time later....

The performance of AmigaBASIC is reasonable for the "bread and butter" functions such as opening windows, throwing shapeless gobs of colour at them, driving the audio channels etc., but the performance of the logic flow of the program, that is, the Mechanics of the Engine, leaves one wanting a lot more. In the early days of the Amiga, as an exercise, I knocked up a "WORM" game in AmigaBASIC, utilising the joystick port, mouse port, keyboard, graphics and menus. I found it painfully slow at times, and had to forsake some of the more attractive aspects of the game to make it acceptable. Also, as the program got larger, parts of the code did not get exercised, but what I didn't know was that if a block of code was not executed, it was not checked for syntax errors. So how can you be sure you've exercised every line of code in a large application? You can't.

In time, I couldn't stand the product. I realised eventually that Kernel (NOT KERNAL) routines could be called from AmigaBASIC, and also noticed that a compiler arrived called AC/BASIC. These did somewhat redeem AmigaBASIC in my eyes, and I will deal with these aspects later, in a comparison with True BASIC.

I first espied True BASIC at good ol' Maxwell's, that venerable font of hospitality. I studied the manuals, and tried to glean from them what True BASIC was, and why Kemeny and Kurtz, the developers of the original BASIC, wanted to write a new BASIC. I test drove it for while, and liked what I saw. I liked it enough to buy it, hoping that once I got it home and studied it more closely, it wouldn't develop ugly blemishes; I'm sure you know that feeling. It proved to be a winner, a dark horse, and I was soon back in the affray of Amiga programming.

So why do I like it? Allow me to list its main features with particular reference to my experiences:-

Portability - True BASIC runs on Amiga, Atari, Macintosh, and IBM without any modifications, even when using advanced graphics. Obviously a 32 color program on the Amiga will run as a 4 color program on an IBM clown, but it will run. I've verified this with one of my larger applications. I ported the source with DOS-2-DOS to a Toshiba laptop: it ran like a dream. Of course, if the program is stand-alone, it needs to be re-compiled.

Structure - It supports modular programming techniques, with Functions, Subroutines, IF-THEN-ELSE, SELECT-CASE, DO-WHILE, DO-UNTIL, etc., while still supporting the old features like line numbers with GOTO or GOSUB (which I avoid like the plague). It does not support labels.

Graphics - All the graphics functions are present, including grabbing subsets of the screen image into string variables. One interesting feature is the Picture facility, a user-definable subroutine which can transform any plot sequence via standard geometric transformations. Points, lines, circles, and text are drawn in terms of user definable screen co-ordinates. Multiple logical windows are also supported. On advanced machines (read "not IBM") animation and color displays are supported. The animation supported is quite remarkable, where frames are quickly fetched from RAM for realtime animation. The user screen co-ordinates are very useful: for example, if you insist on doing your graphics by pixels, simply define the screen as 0 to 320 by 0 to 200 (lo-res). For compatibility with standard text, define the window as 1 to 25 by 1 to 80.

Memory - True BASIC lets you use all the available memory, with strings of up to 1 million characters allowed. Memory management is reliable and efficient, and fast ram is intelligently used.

Speed - They claim True BASIC is "considerably faster" than interpretive BASICs. This is a real understatement. True BASIC is not interpretive, but is compiled into an intermediate B-code (which is hardware independent). For small programs, compiling is instantaneous, while for large programs of thousands of lines, it can take a minute or so. How does the speed compare with AmigaBASIC? It varies according to the mix of functions and activities, but for such constructs as tight "FOR" loops its up to 50 times faster. At its very worst, or AmigaBASIC's very best, several times faster.

External Programs - You can store functions and subroutines in user defined libraries, and call them from your mainline program. These routines are compiled, take up little space and are quick to include.

ANSI Standard - True BASIC conforms to a proposed American National Standards Institute standard for BASIC.

Interface - The interface is easy to use, with program editing, compiling and execution all done within the same environment. There is also an online Help system, which can be modified or added to at will. The editor is excellent. Scrolling is very fast, full use is made of the mouse, and many advanced editing functions are available such as you'd find on a "basic" word processor. Two features stand out; one is the "do" function, which allows one to point to an external True BASIC program, several of which are supplied, which can process your source as it sits in the editor's buffer, for such purposes as formatting, indenting, checking keywords, expanding user mnemonics etc.; the other is the output windows. There is a list window, wherein one can scroll output forward and backward, the other is a command window, wherein one can, after the program is finished, or during break points, inspect, print or alter variable values, inspect, open and close files, and more, again with forward and backward scrolling. All mouse and menu activities have equivalent commands in the command window; very important if you suffer from rodentophobia, or have arrived at the amiga from an IBM clown environment.

Errors - When it comes to reporting compile or runtime errors, the response is quick, the explanation lucid, and the cursor is perched expertly at the site of the problem. The compiler's objection appears in a little reserved strip at the bottom of the window and remains until the next error or warning. One useful feature is the ability to trap all errors that True BASIC issues, thereby interpreting them and acting on them. It provides you with both the error code and the message. The programmer can also generate his/her own fatal error messages and codes at leisure.

Power - all the numerical and string functions available in AmigaBASIC are present, plus a few extra ones, the most remarkable being extensive matrix support, including inversions, determinants, and matrix I/O. Arrays can be explicitly or automatically redimensioned. The mouse is supported, but only on the left button, since menus are not part of the standard True BASIC. Also, a file requestor with directory search and scrollbar is available to any program.

Variables - True BASIC makes no distinction between single precision, double precision, and integer. Conversion is automatic. Trigonometric functions have 10 digit accuracy, while all other functions have 14 digit accuracy.

Odds and Ends - Maximum array dimensions: 255. Maximum files open at once: 10. Maximum record size: 16.8 million bytes. Maximum file size: 4 trillion bytes (!?). Maximum length of variable names: 31. Maximum string length: 1 million characters. Maximum positive number: 1.8e+308. Minimum positive number: 5.6e-309.

What it lacks - Most of the Amiga specific functions. You cannot open a window with a title-bar, drag-bar etc., nor can you drive the audio channels except as a poor imitation of an IBM clown. This is not to say you can't use all the graphics abilities of the Amiga; that is still possible. But this is not the last word...

What else they offer - Firstly, the Developer's Toolkit. This disk has all the kernel calls coded as True BASIC calls, so any system function can be driven. I have used the developer's toolkit to successfully drive the serial port, load diskfonts, write text via the kernel routines, open screens and windows, load IFF ILBM files and much more. Of course, you will need the ROM KERNEL manual(s). They also provide some high level True BASIC routines which, in turn, call the KERNEL routines, which facilitate MENU programming, DISK and DIRECTORY access, and animation with ANIMOBS. Secondly, the Runtime Package, which allows you to create stand-alone programs which can be cranked up via the CLI or workbench. Thirdly, several portable disks, written in True BASIC, which will work on any machine: Advanced string library, which offers pattern matching, expression scanning, parsing, text manipulation, and includes a 38,000 word, uncompressed, public domain dictionary, plus a compress program for it. 3D Graphics Library, which provides high level routines for perspective and parallel projections. Sorting and Searching Library, fast code for heapsorts, quicksorts and multi-key sorts. The source is provided with these three packages, so you may tinker to your heart's desire, or learn a little about the subject in question.

A comparison - Some of you may have spotted on the 1.2 extras disk, some AmigaBASIC code for reading ILBM IFF files, and converting them to ACBM format. The program was awfully slow, but is was cute, so I thought I'd port it to True BASIC. Along the way, I managed to eliminate every call to the "kernel", except one: find the current raster port address. Result? For an average low resolution IFF ILBM image, excluding the time reading the data from the disk, AmigaBASIC took 105 seconds, while True BASIC took 12 seconds. So I compiled the AmigaBASIC program using AC/BASIC. 38 seconds. For an average high resolution image, AmigaBASIC took 360 seconds, compiled AC/BASIC took 170 seconds, and True BASIC took 42 seconds. Next time, I'll try using kernel routines at every possible juncture in the True BASIC program. And while I'm talking about compiled AmigaBASIC, it has a feature I dislike: your L: directory must contain all the AC/BASIC support modules, and relevant BMAPS must be available. A runtime True BASIC program is completely self-contained. For the above IFF programs, the compiled BASIC, when fully loaded, occupied 168K of RAM, while True BASIC occupied 80K.

The True BASIC manuals - The reference manual is the same for all computer brands, since the language is portable. There is also a user's guide, which is specific to your brand of computer. It takes you through all the facilities of True BASIC with special reference to your machine and it's particular user interface. Each manual is about 300 pages in length with ample demonstrations, examples and pictures.

The future - True BASIC release 2.0 is available for the IBM, and is therefore coming to the Amiga, if portability is to be preserved. New features are: Modules. Currently, external routines, unlike internal ones, cannot automatically share variables with other routines, or with the calling program. It must all be via the call parameters. With Modules, routines can, via PUBLIC, PRIVATE and SHARE keywords, share variables with other routines, whether they be internal, external, or within the same Module. Preloading external libraries is supported, thus expediting the compilation or execution of programs which call external routines. There are extra menu options and commands, such as the TRACE facility, which allows variable-speed line-by-line tracing. Numerical co-processor support is there, although this is no guarantee it will be supported on the Amiga. One other thing: Inovatronics have developed a True BASIC support module for Power Windows 2.0.

Before closing, I would like to present you with some sample code. The code writes some text onto an interlaced high resolution screen with 4 colors, using the Kernel text routines instead of the True BASIC PRINT or PLOT TEXT functions, so as to achieve the best possible display speed. The libraries referenced are provided by True BASIC:-

```
LIBRARY "Toolkit:libs/graphics*"
LIBRARY "Toolkit:libs/amiga*"

DECLARE DEF Move,Text,Addr,Nullt$
DECLARE DEF SetSoftStyle
DECLARE DEF SetDrMd,CurrentRPort

SET MODE "LACEHIGH4"

LET rp = CurrentRPort

 ! Write 42 lines of "I Love Lucy".
 ! "R" can be checked at each stage
 ! for valid return codes.

LET T$ = "I Love Lucy"
LET R = SetDrMd(rp,255)
LET R = SetSoftStyle(rp,2,255)

FOR PosY = 1 to 42
 LET R = Move(rp,0,PosY*8)
 LET R = Text(rp,Addr(Nullt$(T$)),len(T$))
NEXT PosY

END
```

Then choose "RUN" from the menu, and the above text appears at a dazzling speed.

External programs written in such languages as "C" or assembler can be called from True BASIC, but do require a simple modification after the link, which a program is provided for. I have successfully written special purpose assembler routines for True BASIC.

Price - True BASIC was, last time I looked, $US99.95, as was the runtime package, but if you bought them together, it was $US149.95 for the pair. The other three disks mentioned were $US49.95 each. Which, times 1.4 for the exchange rate, comes to $140, $210, and $70 respectively. Add 20% if you're unlucky, and it's $168, $252, and $84. Otherwise, Maxwell's have all the disks except the runtime package. And I think they're reduced because it isn't selling. Ignoti Nulla Cupido.

In conclusion, I would like to declare that I'm not trying to sell True BASIC to you. Neither am I trying to sway you from using "C", Modula II, assembler or Forth (what a language!). These languages all have their place. I cannot imagine writing Sculpt-3D in True BASIC; Sculpt-3D needs to squeeze all it can out of the 68000 processor. And if the size of the executable module is critical, the 70K extra you get with the True BASIC runtime shell may be too much for you. Of course, when CSA release their 68030/68882 board, True BASIC is going to look like today's assembler, and then we can have the best of both worlds. For many programmers, especially the ones that are recreational programmers, True BASIC is well worth a look. But be sure to buy what you like, if you ever buy anything at all. All I hope to do by this article is what Amiga users have been doing for a long time: bring a good product to the attention of more people, simply because it deserves more recognition, and could in turn reward you with greater pleasure or productivity.

It seems I've run out of space and time. Sorry, but I'll have to leave the proof that I'm a real man till next time.

I produced this article entirely on an AMIGA, using WORDPERFECT, with absolutely no hardship or anguish. Spelling check took about a minute, with the help of FaccII. I will admit, though, that WP has bugs. I look forward to an upgrade.

The address and phone number of True BASIC is/are:-

True Basic Inc.
39 South Main St.
Hanover NH 03755 USA
Phone (from Australia) 0011 1 603 643 3882

A $95 Modem Kit
by George Wahr

Like a lot of other people, I started out modeming at 300 baud. At first, I thought it was God's gift to the computing masses. Now I could talk to other computers, download software, send messages anywhere in the world, even upload school assignments onto the RMIT Cyber. For a while I couldn't be happier. Then I saw a 1200 baud autodial, auto everything modem. No more constant dialling up engaged bulletin boards - the computer does it all for you. No more slow downloads - what used to take me half an hour would take only seven minutes at 1200 baud. With 300 baud, the most you could download in 45 minutes was about 55k - anything bigger you could forget about.

I had been used to the slow 300 baud screen refresh, where if you accidently pressed the wrong key, you almost had enough time to make a cup of coffee until it was ready again. But when I saw 1200 baud screen refresh at 4 times the speed I had to have one!

The problem, of course, was the cost. With a whiz bang all bells and whistles modem costing around $450 - $500, I'm afraid I had to put up with my old 300 baud modem for quite a while yet. Unfortunately, though, the decision was made for me when the main BBS that I logged onto changed its policy and stopped 300 baud access.

It was time for serious action. I shopped around for a while, then spotted an advert in "Your Computer" for a 300 - 1200/75 modem kit for only $95.00! At the time I saw the advert, this wasn't something I could take advantage of since no split baud rate (1200/75) software existed for the Amiga. (1200/75 allows the Amiga to receive at 1200 baud, and send at 75 baud. As you only send usually a few keystrokes to execute a command on the host computer, it is almost as good as 1200/1200 baud. If you want to upload a file, however, it would be better to redial and do it at 300 baud, since 75 baud is four times slower!) Once the public domain program **vt 100.sb** became available, I was ready to go.

The solution now seemed clear - buy a kit for $95, con a friend into helping me put it together, download the 1200/75 software off the board at 300 baud and fire the system up, then say hello to the world of 1200 baud. And except for a few small problems that's what happened.

Being a pessimist, I decided to go and check the modem out, so a after a quick phone call to ABE computers in Burwood, I jumped in the car and headed East. When I enquired about the $95 modem kit, Max (the owner) rubbed his hands and pointed me to the front of the shop where behind a dirty window and half an inch of dust, he had a kit on display. A request to see a working model was met with "if you wanted a complete one, why buy a kit?" I held my ground and finally, after much rummaging around and cursing, a completed modem was produced.

Okay, I said, let's see it work. A stunned look appeared on his face. Again I stood my ground and with more cursing, he headed over to a PC clone and plugged it in, but after a search of the whole shop, he told me he did not have any comms software because the kid he employed on the weekends must have pirated it.

After much debate and a $200 deposit, he let me take the demo model home to evaluate. I was skeptical, but to my surprise it worked well. The screen flashed at 1200 baud, downloaded properly, everything worked perfectly. I was getting one and that was it.

The kit comes with photocopied instructions, and led by a techo friend we whacked the kit together in about two hours with only one small problem - a little 30 cent transistor was missing. I had to wait all weekend to get another one. Once connected and without even putting it in a case (which I still haven't bothered to do!), I plugged it in and it has worked perfectly ever since.

The only real disadvantages are that not all bulletin boards support 1200/75 (but most of the big ones do), and that when you upload it has to be done slowly at 300 baud. For me, though, saving a few hundred dollars makes up for it. I'm laughing all the way to the bank. I recommend this kit, it's not hard to put together and if you do have trouble, Max will finish it for you for around $40.

Ratpad
by Peter kinross

Most people with Amigas use a mousepad. It makes sense not to scratch the bench top, and they sure make it easier to slide the mouse around. But mousepads usually cost about $20, and I thought that for that much I would just keep putting up with the scratches etc. That is until I actually tried one. When my neighbour purchased his Amiga, he got a mousepad too; well the difference is just too much, they really do make mouse use a lot easier. So I decided to get one.

However the thought kept going through my mind that if I got a large one the keyboard could fit on top and when the kids wanted to play games they could have a large surface to move the mouse about on. But if 5" x 5" costs $20, what would one 24" x 24" cost? (pro rata, that works out to be over $460!).

Several phone calls and I found a place in Braeside that would sell me a piece for under $20. He even gave me choice of color.

It's great, no matter where I move the mouse it glides beautifully! The only catch is, I had to make a support to bridge between the keyboard legs, as they would have left depressions in my new ratpad.

The factory I went to is at 206 - 212 Governor Rd. Braeside, tel 587 1825. The owner was very pleasant and quite co-operative.

An Answer to Last Month's AmigaBasic Question?
by Peter Kinross

Last month, Mark Kelly posed a question concerning cutting and pasting in Basic (the Basic referred to here is AmigaBasic). His problem occurred in using the right Amiga and X combination to cut text that could subsequently be repasted. When he used this feature he frequently lost his text. It wasn't there when he went to repaste it.

In Basic, cutting, copying etc are easily achieved. With the mouse, simply select the beginning of the required portion of text, and with button still pressed, move to the end of the portion you want to select. Release the mouse button, and the selected piece will be highlighted. Then hit the correct key combination. Amiga-x to cut, Amiga-p to paste, or Amiga-c to copy. Nothing could be easier, - or could it?

Basic also has another editing feature - you can replace any piece of selected text (highlighted text) with any new piece of text simply by typing while the old text is highlighted.

What Mark may have been doing is pressing the x key before pressing the right Amiga key. Result: his selected text is replaced by the new text, ie `x'. If that's the problem, the answer is simple - make sure you hit the Amiga key before you hit the x key.

Dear Editor

I sympathise with, and understand your concern (anguish?) over lack of contributions for the newsletter and that raises such feelings of guilt that I am prompted to write.

I purchased my AMIGA 500 in October 1987 after having had about 6 years of very satisfactory computing with a TANDY TRS 80C. Though I am used to writing at assembly level for a variety of Motorola 8 bit machines in my work, I chose to do most of my home programming in BASIC on the TRS 80C as it was a convenient environment and fast enough for most of my applications. I developed several programs to assist our local swimming clubs and produced many issues of newsletters for clubs of various types on my mighty little CoCo. The greatest single attribute I found for that durable and often misunderstood machine was its reliability (both with hardware and software). Seven years operation and not a hint of trouble. My experience was not isolated, many friends had similar reliability from their 80Cs.

The upgrade to the AMIGA was an exciting prospect. No more clunky 8 bit speeds, real graphics, the prospect of a very wide range of exotic software support, the promise of the ability to easily program in C (after all the operating system was UNIX-like, I was told). It all added up to an impression of the perfect home computer.

Oh dear, what a come down!! Firstly, the reliability could be kindly described as poor. My disk drive in my first 500 was so sick that that the supplier (ULTRAPHASE) replaced the whole Amiga without question or fuss. My second one has already fractured a wire in its mouse cable (a fairly common complaint from what I observe), one mouse button is "dodgy", and I have repaired the joystick 3 times already. The power supply also appears to have suffered a fractured wire in its cable as the whole computer dies from time to time when the cable is jolted. By comparison with my old TANDY TRS 80C, much of the software seems "buggy", the operating system is harder to drive, and the documentation unbelievably scant.

Mind you, I was really very niave to have been carried away by the euphoria of it all. I should have realised that the AMIGA was really just another computer, even if vastly more complex than my old TRS 80C. So I am prepared be patient with the operating system whilst it slowly reveals its power and secrets (though, I suspect, not without a fight), and I can be understanding of clever software that invites users to push to the limits such that it continually lives on the brink of a disastrous crash (Gomf where are you?) - but what really galls is to find that all this sophisticated hardware and software is let down by Commodore's apparent refusal to pay anything more than lip service to a real quality control program during manufacture. No computer can be considered first class if it cannot be relied upon to function 100% every time it is switched on. I know my experience is not isolated. Lets hear it then from other users who have had reliabilty troubles and see if some pressure can be brought to bear on Commodore to equal the quality standards matched by other manufacturers.

Enough axes, now for some bouquets. Firstly, the WORKBENCH magazine is a real treat. Professional in style and layout, full of useful information, and the easy to read format makes my yearly subscription good value - even though I cannot get the extra benefit of attending meetings. Currently, I am rewriting some old TRS 80C swimming club software to run on the AMIGA - still in BASIC at the moment but I hope soon to do it in C. However when it's ready I will write and share it with your readers (PD, of course) unless this first letter has offended too many of your readers. Just to whet appetites, here is a brief description.

The program suite is called Swimming Lane Allocation Program (SLAP). The purpose is to accept entries for a swimming carnival and place the swimmers into heats according to SWIMVIC seeding rules. The seeded heats are combined with titleholder details, record information, event titles and printed to make an acceptable Program for the carnival. Designed to cater for up to 99 events per carnival, 20 heats per event, up to 8 lanes per heat, and as many as entries as will fit onto one disk, it can cope with most one day carnivals with ease. The first translation will not exploit the AMIGA fancy bits of course, but if I can be convinced they will add some real value to the functioning of the program, I may include them in the next revision. The real goal, however, will be to perform a rewrite in C both for reasons of speed and as a self education (flagellation?) process.

I end this letter with a plea for some help. The AMIGA BASIC manual discusses using libraries for some very clever functions and the demo programs on the EXTRAS disc included some examples. But where can I obtain a listing of the library modules, descriptions of their functions and examples of how to use them? Where can I obtain descriptions of the file formats for bit images (ACBM, ILBM, etc)? Where do I find out about INTUITION, windows, screens, gadgets and how to use them from within BASIC or C. The bibliography in the back of my AMIGA manual gives a whole range of books that MIGHT be helpful, but where can I see reviews by people who have experienced using these books? And why is this information NOT supplied when I purchase the AMIGA in the first place? I look forward to enjoying the continuing high standard of your magazine and remain,

Yours Sincerely,
Sandy Gray

===

Advanced Graphics SIG

by Geoff Holden, the Reluctant Convener

Meeting 3 - February '88

One of the hazards of SIGs, I am rapidly finding, is that the joker who stands up front during meetings finishes up doing most of the work. As the joker in question, who had heretofore elevated laziness into an art-form, this is distressing. Also, as I seemed to do most of the talking, this report of the last SIG group will sound as self-centred as the average diary entry. Well, perhaps not QUITE that bad. Warnings over.

I had laboured long and hard to create a couple of programs to demonstrate anti-aliasing algorithms. Yes, they were in BASIC, but it is still possible to emit cognitive sweat while working in that language. Jetson, stop laughing ! This is serious. Anyway, I had begged our beloved Editor to use his machine to demonstrate at least two different approaches to anti-aliasing (which is smoothing out the jaggies in drawn lines). He agreed, with what in retrospect was a sinister laugh. Little did I realise that his machine has been Jetsonized, and will only accept programs in socially approved languages (i.e. C, C and nothing but C). I could not get my disc to load - although it was fine on returning home to my machine. So I had to resort to handwaving to explain what I was on about. This is not unusual - what is strange is having an audience...

Briefly speaking, anti-aliasing is done by plotting pixels of half the line brightness intensity at spots where there is a rapid discontinuity, so that the line looks like this:

```
        .XX
       .X.
      XX.
     .X.
    XX
```

I have discovered that if you draw a line at x - 2 at intensity of .5, then another line at x + 4, at an intensity of .5, and finally at x at full intensity, then you get a reasonable quick and dirty bit of dejagging. I have not seen this in the literature, so it may be original. I will accept your worship and adoration.

The other method is by Fujimoto and Iwata (IEEE Computer Graphics and Applications, December 1983, pps. 26-34). The algorithm as printed doesn't seem to either work or make much sense, either. I am working on it.

Anyway, the point behind my getting into a lather on this is that anti-aliasing algorithms are not only important in dejagging images, but also provide a way of rapidly moving through voxel space. That is, if your object space is made up of millions of little volume elements (like the cubelets in a Rubik's cube), then for raytracing, you have to fire lines through that space. And you use a three-dimensional anti-aliasing algorithm to do it. (Not just a line-drawing algorithm, because you must be certain to include EVERY voxel even grazed by the line.)

Whatever, I bored them witless on this theme for a while, and then Andrew Dunbar woke them all up by showing some splendid raytracing images he had created using DBWRender, a PD program. I wonder why there is such an obsession with balls in raytracing...

Sensing that the group was enjoying itself, I then launched into a particularly inept demonstration of "Sculpt-3D". Despite what it looked like, it really is a lovely program. And I hear that version 1.1 is 65% faster. (One sales rep from Ultra said that they were up to version 1.4).

Around this time, a maths enthusiast whose name I didn't catch, actually offered to talk to anyone on the math intricacies of computer graphics. If he turns up again, then I will not only publish his name but ALSO HIS PHONE NUMBER.

Meeting 4 - March '88

This meeting was even MORE Holden-centric than the last. Mainly because once I get up the front and start talking, only the total breakdown of civilisation as we know it will stop me. It's all your own fault...

So, I demonstrated "ROT", the cute little PD program on Fish disk 71. This enables the user to generate 24 frames of hidden-line 3D solids relatively slowly (2-3 seconds per frame for about 50 polygons) and then play them back at video speed. The user-input for creating the 3D objects is through a three window view system which is very similar to the "Sculpt-3d" TriView. Having created an object (or objects), it can then be translated (moved around) and rotated in x,y,z dimensions, and the frames generated. The movement can be made very smooth, and the movement path saved and used for other objects. Main disadvantages are that the created objects are composed of flat, unshaded polygons, and the viewing area of the finished moving objects only occupies about half the screen.

Then, I rattled on about the technique of "octree encoding". This is a method for describing and storing 3D information. Its advantages are:- efficient use of memory, very quick display and rendering algorithms, ability to intersect objects and perform Boolean operations on both single objects and combinations of objects and its ability to interface fairly neatly with raytracing and other procedures. Its disadvantages are that it doesn't take to curves very readily (they take up LOTS of memory) and that your stored data has a finite resolution - that is if you zoom in too close then you see lots of "jaggies".

Broadly speaking, octrees are an alternative to CSG (Constructive Solid Geometry) and Sweep Generation. In the former, you distort primitive geometric shapes and construct your objects out of them, in the latter you sweep out volumes of space by mathematical graphs rotating around some axis. Well, roughly. In octrees, you divide your object space into 8 equal voxels, forming a cube. Each voxel is then declared either completely full, empty or partly full - F,E or P. The P voxels are then divided into 8 and declated F,E or P. And so on recursively, until either you run out of memory or some declared resolution figure is met. So you finish up with a strictly defined hierarchical tree of octanary voxels, each voxel getting smaller as you go down the tree. Now, using a cute numbering system, it is possible to search the whole tree VERY QUICKLY - for rendering, for example. And, as the observer position is defined with respect to the object space, ie how the observer is looking at the topmost (root) octree (the biggie), then display is a matter of starting at the smallest voxel in the furthest corner and drawing the voxels in logical sequence coming towards the observer. Hidden surface is thus automatic.

Now, that is as clear a description as I can do without lots of diagrams and handwaving. The relation of the above and raytracing I hope to clarify at the next SIG. (I have Easter to make up something plausible). But the octree subject is still a hot one in the professional graphics literature - I flourished a bulky file of reprints on it at the SIG, and there are still more coming. So buckle down, fellows - it is conceptually tricky - when you try to work your way through some aspects of it, then, like me, you may be conscious of a throbbing in the temples - but it gives a new way of looking at the object space. And that is where we live.

Next time, if the gods are merciful, Stephen Newnham will demonstrate an image generation program, Enno Davids will be benign and omniscient and I will have laryngitis. So there.

===

The New Lattice C

by Roland Seidel

I've had the opportunity of playing with the latest version of Lattice C, version 4.0, and can now offer a brief and quite unprofessional review and let you know of a great deal on offer for Amiga Users Group members. I am not a professional programmer - more of a reasonably competent fiddler - and it would be good if someone more qualified could offer to do a more thorough review, but here is what I found.

The Lattice agents here are looking for a bit of Amiga publicity and agreed after a some chit-chat to give User's Group members a 30% discount on the developers' package. They would rather sell the whole package but agreed to offer some discount on the compiler as a separate item.

Fagan Microprocessor Systems Pty. Ltd.
95 Canterbury Road
Middle Park 3206
phone 699 9899

The Deal

The Developer's Package:

* Lattice C Version 4 Compiler
* Lattice Make (LMK)
* Text Management Utilities (TMU)
* Lattice Screen Editor (LSE)
* Metascope Debugger

List price	$595.00	($678.30 inc tax)
AUG price	$416.50	($499.80 inc tax)

Just the compiler:

List price	$245.00	($279.30 inc tax)
AUG price	$196.00	($230.30 inc tax)

FMS more commonly handle PC developers' software ("Tools for the Professional") and as Lattice agents are adding Amigas to their list of supported machines. They gave us a perpetual catalog of all their software, which I will leave with the AUG library - for when they get more Amiga stuff or if anyone out there wants to see the professional stuff available on PC's. It was refreshing to find a shop with more than just salespeople in it, meaning that they offered pretty reliable technical support.

Well, I read all the documentation and said to myself, "Yeah, it looks like they have done some work but I still remember waiting two minutes for 'Hello, World' to compile with an earlier version". I hadn't used Lattice since someone showed me the speedy Aztec Compiler, so I decided to put them both through the most rigorous test I could think of... lets try 'Hello, World' again.

Aztec compiled it in 53 seconds and Lattice did it in 45! Shock horror! I tried a few other things; windows, gadgets, counts ... just mucking around really ...., and Lattice invariably came up somewhat faster! I don't know what claims to speed they make but I definitely found compilation faster.

I forgot to check code size but the documentation made mention of smaller code a few times and some of the changes seemed specifically aimed at achieving that.

A new set of header files called "Proto" manage to enable direct calling of Amiga libraries without going through the stubs in Amiga.lib. You have to set up references to functions using a new # line involving reference to the routine and the registers it uses for input and output ... eg:

```
#pragma libcall DOSBase Open 1E 2102
```

It looks a bit arcane and it feels funny using CPU registers in C but you don't have to use them; they're there if you need them.

A number of functions have been built in and referenced by, for example, _builtin_printf(). All header files come in compressed and uncompressed format to relieve disk space problems. Compiling and linking can be done in a single, simple line:

```
LC -Lhello
```

or you can use the Make facility that seems as good as any.

I'm not sure I liked the Screen Editor as much as Z, but we are such creatures of habit. With all the utilities designed to talk to each other and to produce just the sort of output the others need to speed things up, it would be an advantage to get the whole package and use it.

The best bit, I thought, although I suppose it has been there all the time and I just didn't know, is that you can get cross compilers for PC's, Macs, etc. I intend doing some stuff for schools and it is nice to know that I don't have to rely on the school having an Amiga or to resort to carting mine around.

I liked it.

==

Using PatternMaster Pattern Files In AmigaBASIC Programs

by Alan Garner

Background

When I bought my Amiga in December, 86 I became interested in the use of the AmigaBASIC PATTERN function to create fill patterns for graphics programs. I soon found that it was extremely tedious to sit down and calculate the integer value for each line of the pattern I wished to create. So, I decided to write a program called PatternMaster which handled all the calculations for me. This worked fine, but I soon became bored with patterns that consisted only of two colours (the current foreground and background colours).

It wasn't until I read an article by Bryan Catley in Amazing Computing that I realized AmigaBASIC could create multicoloured fill patterns. I quickly upgraded the PatternMaster program and compiled it using the AC - BASIC compiler. The current version (v.3) of the program is now available from the AUG software library.

About this program

When PatternMaster saves a pattern file, it saves it as a 320 byte file (160 integer elements). These 160 elements represent the 5 bitplane, 32 line pattern with each bitplane saved in its entirety before the next. Members of BaSIG who have seen PatternMaster running have asked me how the pattern files could be loaded for a screen with a different depth or to create a pattern with a different line length (2, 4, 8, or 16 lines). This program does exactly that. You input the pattern name, the number of bitplanes and the pattern length you require and it does the rest.

Hopefully the remarks in the program will be self - explanatory, but for those who want to know more I suggest you read Bryan Catley's article in Amazing Computing Vol.2, No.9 Pp.76 - 79.

```
' ==========> Program initialization and User input

CLEAR : DEFINT a - z : OPTION BASE 1

patternfile$ = ""
WHILE patternfile$ = ""
      LOCATE 2
      PRINT "Name of pattern you wish to load: ";
      LINE INPUT patternfile$
WEND

bitplane = 0
WHILE bitplane < 1 OR bitplane > 5
      LOCATE 4
      PRINT "How many bitplanes do you require (1 - 5) ";
      INPUT bitplane
WEND

linepower = 0
WHILE linepower < 1 OR linepower > 5
      LOCATE 6
      PRINT "What is the power of the pattern height"
      PRINT "(eg. 4 indicates a height of 16 lines) ";
      INPUT linepower
WEND

' ==========> From the parameters entered we determine
' the pattern height, the maximum number of colours,
' and the dimension for the pattern array.

' (The pattern array must be dimensioned to the
' power of 2 greater than the number of
' elements in the array.)

patternheight = 2 ^ linepower
maxcolour = (2 ^ bitplane) - 1
elements = bitplane * patternheight
count = 0 : arraylimit = 0
WHILE arraylimit < elements
      count = count + 1
      arraylimit = 2 ^ count
WEND
DIM patt%(arraylimit)

' ==========> Initialize required screen and window.

title$ = STR$(bitplane) + " bitplane"
IF bitplane > 1 THEN title$ = title$ + "s"
title$ = title$ + " and " + STR$(patternheight) + " lines."
SCREEN 2, 320, 200, bitplane, 1
WINDOW 2, title$, , 0, 2

' ==========> Initialize palette with PatternMaster
' default palette.

RESTORE PaletteData
FOR a = 0 TO maxcolour
      READ r, g, b
      PALETTE a, r/15, g/15, b/15
NEXT
```

```
PaletteData:
DATA  0, 0, 0, 4, 4, 4, 7, 7, 7,10,10,10
DATA 15,15,15, 8, 0, 5, 8, 0, 0,11, 0, 0
DATA 15, 4, 5,15, 0, 8,15, 7, 9,15, 9, 7
DATA 15, 7, 0,15, 4, 0, 6, 4, 1, 8, 5, 0
DATA  9, 7, 0,15,10, 0,15,15, 0,15,15, 6
DATA  9,15, 0, 0,10, 0, 0, 7, 0, 0, 8, 6
DATA  0,13, 6, 0,15,11, 0, 9,15, 0, 5,15
DATA  0, 0, 9, 4, 3, 6, 6, 0, 7,10, 0,15

' ==========> Load required pattern file.

OPEN patternfile$ AS #1 LEN = 2
FIELD #1, 2 AS integer$
count = 0
FOR a = 0 TO bitplane - 1
      FOR b = 1 TO patternheight
            recordnumber = (a * 32) + b
            GET 1, recordnumber : count = count + 1
            patt%(count) = CVI(integer$)
      NEXT
NEXT
CLOSE #1

' ==========> To use a multicoloured pattern we must
' issue a sequence of commands.

' 1.  Use the PATTERN command to define the pattern
' 2.  Change the 1 byte areafill pattern size pointer
' in the current rastport structure to indicate a
' multicoloured pattern with 2^linepower lines.
' 3.  Set the background and foreground colours to
' the minimum and maximum colours respectively.

PATTERN, patt%
POKE WINDOW(8) + 29, 256 - linepower
COLOR maxcolour, 0
PAINT(10, 10)

' ==========> Reset pattern to full fill and display
' palette

POKEL WINDOW(8) + 8, 0
POKE WINDOW(8) + 29, 0
FOR a = 0 TO maxcolour
      LINE(a * 9, 160) - (9 + a * 9, 190), a, bf
NEXT

' ==========> Wait for any event.

SLEEP : SLEEP
WINDOW CLOSE 2
SCREEN CLOSE 2
```

==

My Amiga Is Alive or There Really Are Bargains Out There Today

by Peter Kinross

Going from my old Z80 Sharp computer to my new Amiga, I thought that there could not ever be any need for more than 500k of memory.

Well, I suppose I am not alone in coming to terms with the well known adage that the work to be done (or enjoyed) on any computer will always expand to require at least 10% more memory than you currently have.

I have written some programs in Amigabasic to assist in my work. One of them ran me out of memory, and it is only half written. On top of that I have been trying to use the spreadsheet program Maxiplan Plus, and that little fellow can really soak up the K's. It has some really amazing features, but it is hungry.

So bullet biting time came and I was faced with the task of attempting to fit in purchasing memory expansion with my wife's funny ideas that we needed carpet instead of concrete in our new extension. The more I looked into the problem, the more exasperated I became. My requirements were simple - 1 meg, auto config, clock and pass through bus. Looking in all the Amiga magazines, American, English and Australian, I was thoroughly discouraged. It seemed strange that memory for the Amiga should cost twice as much as memory for an ibm (whoops, did I miss the capitals?). There remained the option of looking at the offerings presented (sadly not anymore) at our monthly meetings, and at those advertised in this magazine. But there didn't seem to be any available that met my requirements, and that met my non-existent budget.

However at one of the Basic Special Interest Group (BaSIG) gatherings, another sufferer mentioned that he had recently purchased a memory expansion unit from Peter Valkovic out at Hoppers Crossing. So with fear and trepidation I phoned him (749 4128), left a message, and waited for his call. He rang back, described his expansion unit - 1 meg, auto-config, clock with pass through; and his price $520. Well at last, something that fitted all my criteria, except budget, but who needs carpet anyway - concrete wears so well.

I placed the order with him and he delivered the unit to my office within a couple of days. Like a kid with a new toy, a not too unaccurate analogy made by my wife, I raced home plugged it in and ... and .... and -- nothing. Disaster, still the same old 386000 on the screen. I rang Peter and almost cried. Now I knew that as he lived at Hoppers Crossing and as I lived at Patterson Lakes, one of us was in for a long trip, and I sure didn't want it to be me. Well, and this is where the alternative heading comes in, Peter was fabulous, he said that he would call around the following evening and fix it up, which he did. It was only a slightly larger than normal 86 pin plug on my computer, a few strokes with a file and away she went.

Just pause for a moment and think what would have happened if I had brought the memory unit from a dealer. Who would have had to do the running around? Who would have had to wait for repair or replacement? The service I received from Peter was what we all want these days but NEVER see. I hope you understand why I am impressed and thankful.

What a wonderful computer is the Amiga with a bit of memory under its belt. How nice it is to see 1410000 when firing up. And how handy to have enough memory to use a recoverable ram disk.

==

Special Interest Groups

The Amiga Users Group has a large number of special interest groups (SIGs), most of which meet after the main AUG meeting each month. Below is a list of SIGs and their co-ordinators:

Video SIG	Geoff Wood	580 7463
Music SIG	Roland Seidel	890 3934
Beginners SIG	Has guest co-ordinator each month	
Developers SIG	Chris Tremelling	557 1349
Basic SIG	John Elston	375 4142
Games SIG	Luke Devlin	n/a
Communications SIG	Bohdan Ferens (Drac)	792 3918
Beginners C SIG	Mal Woods	288 5472
Advanced C SIG	Eric Salter	861 9117
Desktop Publishing SIG	Details not yet tracked down	
Advanced Graphics	Geoff Holden	n/a

A group of AUG members from the North and North Western suburbs have started a "North Western SIG", and this group meets twice a month in Essendon. Contact the co-ordinator, Neil Beatty on 370 9976 between 7 and 9pm for more details.

A review - Megaboard II

Progressive Peripherals & Software
Price - $849

Distributed locally by Diskworks (Sydney)

When I first bought my Amiga I thought, like most people, that 512K would be plenty of memory for the kind of things that I wanted to do. Sure enough, after only a year, I decided I could not live without extended memory. I had looked at all the options open to me, there was a do-it-yourself hack, which still looks attractive, there were two locally made boards (Proton & Subordinate Systems) and a swag of overseas boards. I eventually narrowed the field down to three boards, Memory Plus (Subordinate Systems), Spirit 1.5 (Internal) & Megaboard II. At the time Memory Plus did not autoconfigure, and that was the only reason that I did not choose that board.

With the field now down to two boards, I decided to bite the bullet and buy the Spirit Board. The reason I decided to buy the Spirit Board was that I did not want an ugly box fixed onto the side of my Amiga (more about that later). The Spirit Board seemed to be the answer to my problems, 1.5 Meg of memory, autoconfigure and has a battery backed clock all for $799.

I rang Diskworks and ordered it, and it was delivered the following day. I started the installation, removing my Amiga's cover, RFI shield, and the 68000. I discovered, to my horror, that the board was not going to fit. I rang Diskworks and they said that some Amiga's have a slightly larger diskdrive case, and that prevents the board from being seated correctly. Their solution was to 'trim' the diskdrive case. I did not want to do this, so I suggested that I return the Spirit Board in exchange for the Megaboard II, having resigned myself to having an ugly box attached to my Amiga. Diskworks agreed although they were under no obligation to do so.

A few days later, the Megaboard II arrived and I quickly opened the carton to find a disk and a small plastic case. I attached it to my Amiga, and I am still amazed how small the board is. The Megaboard II is not a big ugly box as I had originally thought, it has been well designed and the board has been running for a year now, without a single problem.

Dave's Raves

by David Peel

Whenever I speak to other computer minded people, I find it hard to sort their dreams or ideas of what should be from what can be realised. For instance everybody I have spoken to says that it should be very easy to find a simple program to enable me to print pages from Pagesetter sideways using a dot matrix printer. I have tried Flipside and it doesn't do it. In theory it should be possible, as programs like Deluxe Paint and Deluxe Print can print Sideways but to do the straight text work which I want to do for my simple little church notice sheet which is both sides of a folded A4 sheet, surely the answer is available commercially or in the public domain. So if you have dreamt of it or think that you might get it together one day, please don't let me know but if you have actually seen a program which does what I seek please let me know posthaste.

I have been really excited by the offerings in the public domain. One program which I use all the time is called Dux5. It is a late relative in the Dirutil family. It is fairly complicated at first and I am often tempted; and do yield to the temptation to use the simpler Dirutil2. Nevertheless, I do think that I like these programs a lot and would like to encourage those whose minds are far more agile than mine to continue their efforts. There is, however, a difficulty with this shareware business. By the time you send $20 Australian to someone in the USA, it will cost you about $45. I wondered if perhaps the User Group could post a list of the shareware programs most in demand and invite people to make a small contribution and then send the whole amount to the author. Perhaps the committee could give the matter some thought.

Just the other day, horror of horrors, a little message appeared, volume "thingummy" has a read/write error. Never one to let these things upset me, I thought Aha!, I'll pop in Disk-X and see if I can find it. Well, it turned out not to be that simple and neither did Newzap3.0 help much either. With NewZap3.0, you need to know the file name before you can read anything and then you can't correct the checksum. Disked can change the cylinders, but not the sides. Disk-X turned out not to be much use either, being unable to change the checksum. In the end, I used the Disk Error program on Marauder, but then having located the error on Cylinder 3 Side 1 and having been told that it was error 24, header file has bad checksum. No problem, I'll slip in DiskEd and correct the checksum. But DiskEd doesn't enable you to goto the specified side of the cylinder concerned. In the end, I used Diskdoctor and reconstructed the disk from other sources. Unneccessarily complicated, I thought to myself after having seen nice programs for the Commodore-64 like Disector-V which enabled you to analyse and correct errors on a disk. It is, I am told, still a very commercial product. Maybe I'm not sufficiently well read. Maybe there is such a simple, useful, easy to use product for the Amiga.

Book Review

Amiga Assembly Language Programming
by Jake Commander
Tab Books, $26.95

Reviewed by Geoff Holden

There is no shortage of books on 68000 programming. Zaks, Leventhal and Williams have produced them and - given the lovely logicality of the chip - there is more than enough information available for the assembly language neophyte to the advanced programmer. No problem, folks.

BUT if you shape up to the Amiga with one of these books in your hot-programmer hand, you will not even start the game, let alone pass GO. The Amiga is a multitasking machine which means that there are few definite places in the memory where you KNOW things will be found, and the designers have kept themselves loose and flexible as to the locations of even the most essential of inbuilt routines - you know, the basics like "accept a character from the keyboard" or "print a character on screen".

If you are a assembler programmer from wayback, who has polished his neurones on the Z80 or 6502, then this is disturbing. You cannot JSR to $03BF to do this or that. You must open a library file, have an INCLUDE or two, point to a structure, and then pass a parameter or two. Finally you call the subroutine by a name that might not have occurred to you in a nightmare. And, just in case something goes wrong, you must allow for every possible screw up by having graceful exit routines for the eventuality that a library file can't be found or whatever.

Part of this rigmarole comes from the way the Amiga is designed - which is with the C programmer firmly in mind. And part from the desire of the designers to change the operating system in quite dramatic ways at some stage in the future. All well and good. And a good clear explanation of the whole shooting match would bring a smile to my lips and a cheque book into my hand. SO......

The above named book is an attempt at that clear explanation. It is not designed as a stand-alone text - you will still need the Zaks or Leventhal - nor does it replace the ROM-Kernel books. But it does try to start from the beginner level and take you up to the point where simple text handling programs in assembler make sense. And the author (who used to write hilarious articles in the magazine "80-Microcomputing") is friendly and not patronising. And yet, something is missing. The guts of the Amiga-specific stuff is pages 79-90, which deals with the libraries, INCLUDES, and calls and it somehow doesn't quite make sense. To me, anyway. Perhaps if you are a C programmer moving to assembler, there will be no problem, since the Amiga design forces the assembler program to conform to C patterns. But moving from other assemblers to the Amiga requires some very patient explaining. And there is not quite enough.

For the rest, the book is competent enough. There is a good list of 68000 instructions, a reasonable explanation of the wealth of addressing modes, a handy list of library routines (suitable for quick reference only - you'll need the ROM books for details) and an appalling index.

It's cheap, available and the only one in town. (Sounds like me, doesn't it ?). 'Til a better book on the subject comes along, it'll do.

Public Domain Update

Only two new disks on offer this month, from the Amigan collection, put together by the Amigan Apprentice and Journeyman magazine.

Amigan Disk #13

Amoeba Invaders	An extremely good rendition of the classic Space Invaders video game. If you ever liked the original, you'll LOVE this one!
BattleMech	A computer version of the game BattleTech, which tries to capture the strategy, while taking care of the tedious calculations for you
GravAttack	An original game by Steve Peterson. You will have to sharpen your piloting skills to get far in this one. Very good
WBLander	Another Lunar Lander. This one has a neat twist... but you'll have to land to find out! (Make sure your sound is hooked up before you start.).

Amigan Disk #14

A68k	An optimising 68000 assembler. Does a very good job, and includes complete source (for Manx C)
Blink	Just in case, here's version 6.9 of Blink, which was the last PD release of this now famous linker
Compu-Calc!	A neat calculator, not especially feature-loaded, but includes complete source (in assembler)
DirWin	A useful utility, especially if you look at directories a lot. Will automatically bring up the directory for a disk inserted in drive DF0:
DME	An Extra good programmer's editor, allowing remapping the keys (and even the Mouse buttons!) to any other key or even a command sequence. 128

	possible mappings PER KEY make this an EXTREMELY powerful editor. Also allows multiple files in multiple windows, etc
FFormat	A fast disk formatter, used to format the original of this disk. Includes the capability to install your disk immediately after format, thereby replacing both Format and Install in your C: directory
Resident	The resident command, plus several support files. Read the docs to find out uses for this
Riley	Several files from Mark Riley (author of MusiCraft, which later became Sonix). Included are a program packer with a twist (packed files unpack themselves!), an IFF SMUS song player which can be used from CLI or Workbench, and a program to turn off the filter in the Amiga sound section (on the A500 & A2000 only). Also includes some Christmas songs to play with the Player
SClock	I know; "Not another clock program!" This one will do something a little different...it will make sure it stays on the top screen, even custom screens put up by such programs as Shanghai and various terminal programs
SmallLib	A small library which can be used to replace amiga.lib in most cases. Included to help with A68k
SpeedUp	A file from Germany which will change the drive step delay in the trackdisk.device for your drives. Does show noticeable improvement in disk access speeds
Wordwright	A word processor that rivals many being sold for $100 or more! Includes an outliner, mail merge, and multi file capability, just like the commercial jobbies

==

Editor's Column

(written 26-Mar-88)

Hi again. This is another of those columns designed to fill the section of the newsletter between the end of the previous column and the end of the newsletter. Which means, yet again, barely enough contributions to fill the pages of our club journal. Please keep writing about your Amigas. A club isn't a club if **you** don't do your share.

You might have noticed an article or two without a credit for the author. That usually means that the author didn't include his or her name in the article, or at the very least, on the disk label. This newsletter is accumulated from many sources during the month, and it is very easy for the original file or disk, the thing it was mailed in, and the reformatted version on my hard disk to be impossible to be matched up again. So, if you contribute to our newsletter (and I hope you will), don't forget to include your name in the article (at the start or the end, it doesn't matter), and also write it on the disk label, which will help the disk on its way back to you. If I haven't yet returned your disk, I apologise. It'll wing its way back to you soon.

There have been a few questions on what we will or will not publish lately. If you are thinking of writing a review of hardware or software that you've bought, or a book review or instructional article, then your article will probably appear in the next issue of the magazine. The only thing that may hold it up would be if I had to edit the article considerably or re-write parts of it. If, on the other hand, you're interested in writing a news, views, opinions and rumours type column, then you have to be a bit careful with exactly what you say, due to things like libel laws. We can publish fair comment, and your FIRST-HAND experiences, but one or two writers have gone completely overboard on occasion. We also can't publish articles that incite people to break the law.

When you write, please make sure that you save your article in **text-only** mode. This is important, since I probably don't use the same word-processor you do, and all the formatting information and control codes that your word processor puts into your article are just gibberish to me. Text-only means that Ed, the editor that comes with Amigados, will accept it. If Ed can read it, so can I. If you have no way to make your article text only, then submit it anyway, I've written some utilities that may allow me to extract the intelligence from the gibberish.

A few people have asked me for more details about the Wedge hard disk controller interface. The Wedge allows you to attach an IBM hard disk and controller to your Amiga. It is a small adaptor board that plugs into the expansion port on the right side of your Amiga 1000. The Wedge sells for $175 Canadian (special price for members of Amiga users groups), and you can get one from:

Dave Allen
RSI
c/o 9651 Alexandra Road
Richmond, BC
Canada V6X 1C6

Phone (604) 270 0064

Mine has performed faultlessly in the three months I've had it, although the version of the driver software I have won't run two hard disk drives as once. Since it should, there'll probably be a software update sometime.

The Trade Practices Commission recently sent a circular to 2000 computer companies, outlining problem areas with the warranties given with most software packages. I have obtained permission to reprint the circular, and I have arranged for it to be available on our bulletin board, AmigaLink. This document should be read by anyone buying software.

A number of people have asked for details on the AmigaDOS Resident command. There is an article on it in the Amigan Apprentice and Journeyman Jan/Feb '87 issue, available from our club library, and you might also like to look at Amigan disk #14.

Well, that's another newsletter done. We're back to Sunday meetings now, so I'll see you on April 17th.

Newsletter Back Issue Order Form

Issue Numbers:								
Number of issues at $2 each					$			
Less 10% for 5 or more					- $			
Club Use Only					Total $			
Mail to: Amiga Users Group, PO Box 48, Boronia, 3155								
Member's Name:								
Address:								

SOFTWARE ORDER FORM

Disk numbers :							
Disks supplied by Amiga User Group @ $8					$		
Disks supplied by member @ $2					$		
Club Use Only Receipt #: Mailed on: / /					Total $		
Mail to: Amiga Users Group, PO Box 48, Boronia, 3155, Victoria.							
Member's Name:							
Address:							

Application for membership of The Amiga Users Group Inc

Membership is $20 per year. Send your cheque to: **Amiga Users Group Inc, PO Box 48, Boronia, 3155**

Surname: ______________________

First name: ______________________ (no initials)

Address: ______________________

______________________ Postcode: ________

Phone Number: ______________ STD Code: ________

What services would you like AUG to provide: ________

____

____

Signed: ______________ Date: ________

Details on this side are optional

Year of birth: ________ Which model Amiga: ________

Occupation: ______________________

Interests: ______________________

____

____

Dealer's Name: ______________________

Dealer's Address: ______________________

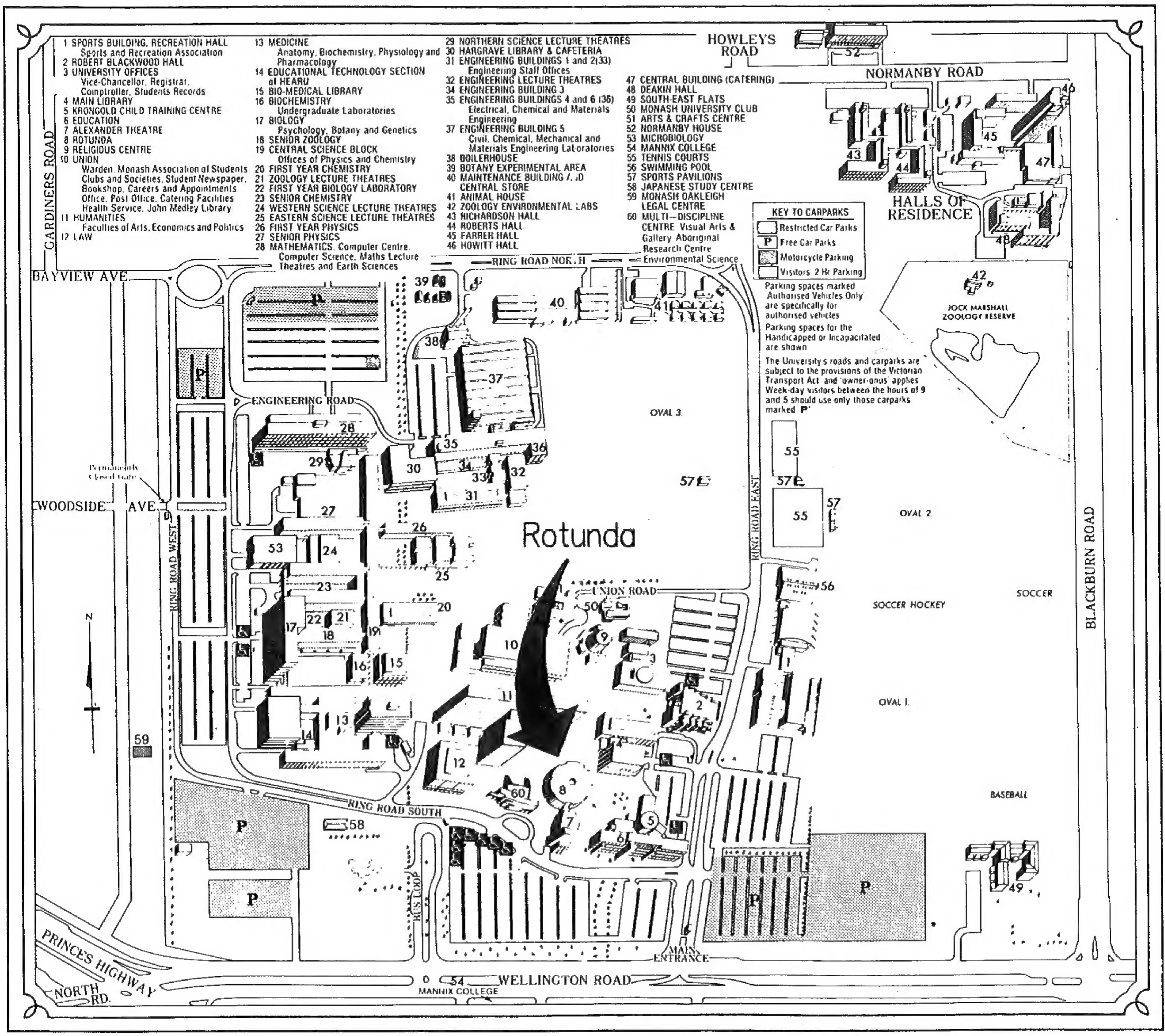
______________________ Postcode: ________

In the event of my admission as a member, I agree to abide by the rules of the Association for the time being in force.

Club Use Only	Date	Paid	Rcpt #	Memb #	Card Sent

AUG now meets on the third Sunday of each month

Monash University is in Wellington Road, Clayton. See Melways Map 70, reference F10. Melways map 84A shows the University Campus in details. I've drawn a huge arrow on the map below to show where the Rotunda is. The best place to park your car is the car park area between Wellington Road and the Rotunda. The entrance to the Rotunda is virtually at the point of the arrow.


1 SPORTS BUILDING, RECREATION HALL
Sports and Recreation Association
2 ROBERT BLACKWOOD HALL
3 UNIVERSITY OFFICES
Vice-Chancellor, Registrar, Comptroller, Students Records
4 MAIN LIBRARY
5 KRONGOLD CHILD TRAINING CENTRE
6 EDUCATION
7 ALEXANDER THEATRE
8 ROTUNDA
9 RELIGIOUS CENTRE
10 UNION
Warden, Monash Association of Students Clubs and Societies, Student Newspaper, Bookshop, Careers and Appointments Office, Post Office, Catering Facilities Health Service, John Medley Library
11 HUMANITIES
Faculties of Arts, Economics and Politics
12 LAW
13 MEDICINE
Anatomy, Biochemistry, Physiology and Pharmacology
14 EDUCATIONAL TECHNOLOGY SECTION of HEARU
15 BIO-MEDICAL LIBRARY
16 BIOCHEMISTRY
Undergraduate Laboratories
17 BIOLOGY
Psychology, Botany and Genetics
18 SENIOR ZOOLOGY
19 CENTRAL SCIENCE BLOCK
Offices of Physics and Chemistry
20 FIRST YEAR CHEMISTRY
21 ZOOLOGY LECTURE THEATRES
22 FIRST YEAR BIOLOGY LABORATORY
23 SENIOR CHEMISTRY
24 WESTERN SCIENCE LECTURE THEATRES
25 EASTERN SCIENCE LECTURE THEATRES
26 FIRST YEAR PHYSICS
27 SENIOR PHYSICS
28 MATHEMATICS, Computer Centre, Computer Science, Maths Lecture Theatres and Earth Sciences
29 NORTHERN SCIENCE LECTURE THEATRES
30 HARGRAVE LIBRARY & CAFETERIA
31 ENGINEERING BUILDINGS 1 and 2(33)
Engineering Staff Offices
32 ENGINEERING LECTURE THEATRES
34 ENGINEERING BUILDING 3
35 ENGINEERING BUILDINGS 4 and 6 (36)
Electrical, Chemical and Materials Engineering
37 ENGINEERING BUILDING 5
Civil, Chemical, Mechanical and Materials Engineering Laboratories
38 BOILERHOUSE
39 BOTANY EXPERIMENTAL AREA
40 MAINTENANCE BUILDING /. .D CENTRAL STORE
41 ANIMAL HOUSE
42 ZOOLOGY ENVIRONMENTAL LABS
43 RICHARDSON HALL
44 ROBERTS HALL
45 FARRER HALL
46 HOWITT HALL
47 CENTRAL BUILDING (CATERING)
48 DEAKIN HALL
49 SOUTH-EAST FLATS
50 MONASH UNIVERSITY CLUB
51 ARTS & CRAFTS CENTRE
52 NORMANBY HOUSE
53 MICROBIOLOGY
54 MANNIX COLLEGE
55 TENNIS COURTS
56 SWIMMING POOL
57 SPORTS PAVILIONS
58 JAPANESE STUDY CENTRE
59 MONASH OAKLEIGH LEGAL CENTRE
60 MULTI—DISCIPLINE CENTRE Visual Arts & Gallery Aboriginal Research Centre Environmental Science
KEY TO CARPARKS
Restricted Car Parks
P Free Car Parks
Motorcycle Parking
Visitors 2 Hr Parking
Parking spaces marked 'Authorised Vehicles Only' are specifically for authorised vehicles
Parking spaces for the Handicapped or Incapacitated are shown
The University's roads and carparks are subject to the provisions of the Victorian Transport Act and 'owner-onus' applies Week-day visitors between the hours of 9 and 5 should use only those carparks marked 'P'
HOWLEYS ROAD
NORMANBY ROAD
HALLS OF RESIDENCE
GARDINERS ROAD
BAYVIEW AVE.
RING ROAD NORTH
ENGINEERING ROAD
JOCK MARSHALL ZOOLOGY RESERVE
OVAL 3
Permanently Closed Gate
WOODSIDE AVE.
RING ROAD WEST
RING ROAD EAST
OVAL 2
Rotunda
UNION ROAD
SOCCER HOCKEY
SOCCER
BLACKBURN ROAD
N
OVAL 1
BASEBALL
RING ROAD SOUTH
BUS LOOP
MAIN ENTRANCE
PRINCES HIGHWAY
NORTH RD.
WELLINGTON ROAD
MANNIX COLLEGE
P

BY PUBLIC TRANSPORT . . . The simplest method is to take a train from Flinders Street or Loop stations on the Dandenong/Pakenham line to either Huntingdale or Clayton. Buses run from these stations to the campus or there is a taxi rank at Clayton. With suitable connections the trip takes about 45 minutes — but it can take longer! An inner neighborhood ticket will take you all the way via Huntingdale station and the bus, but you will need to purchase a comprehensive ticket for the trip via Clayton, which encompasses two neighborhoods. The campus is also served by buses from Box Hill, Blackburn, Belgrave, Chadstone, Jells Park-Glen Waverley, Dandenong-Mulgrave, Oakleigh and Elwood.

FROM THE CITY BY CAR . . . An easy route is along St Kilda Road or Kingsway/ Queens Road and then on to Dandenong Road. The campus's tall Menzies Building comes into view a kilometre or so before the left turn into Wellington Road on which the main entrance is located. Allow 40-50 minutes for the trip. Drivers should note that restrictions apply in some car parks weekdays 9 a.m. to 5 p.m. and fines **do** apply. There is ample unrestricted parking and, closer to buildings, designated two hour visitor car parks — check the map or ask at the Gatehouse.